Theoretical Analysis of Graphcut Textures

Xuejie Qin Yee-Hong Yang
{xuq, yang}@cs.ualberta.ca
Department of Computing Science
University of Alberta

Abstract — Since the paper was published in SIGGRAPH 2003, the graphcut textures has become a popular and important technique for both static and dynamic image texture synthesis. However, the discussion on the theoretic side of the graphcut textures appears lacking with little information provided in the original paper to convince the reader the correctness of the approach. This paper addresses the important theoretic issues related to the technique of graphcut textures using the *complete* $\alpha-$*expansion* graph, which is an extension of the $\alpha-$*expansion* graph. The main contribution of this paper is to correct or clarify some statements in the original graphcut textures paper and to give mathematical support and vigorous proofs for the approach.

Index Terms—Labeling problem, graph cuts, complete $\alpha-$*expansion* moves, graphcut textures.

1 Introduction

Image texture synthesis (i.e. texture synthesis in 2D space) is an active and important research topic in computer graphics. Given an input texture sample, the goal of texture synthesis is to generate a new image whose textures look similar to the input texture. Among the various approaches, the patch-based texture synthesis technique [3, 5, 6] generates a synthesized image (output) by copying small patches from the input image to the output. The challenging problem in this approach is how to remove (or reduce) the visible seams between patches when they are placed into the output. As a popular and important patch-based texture synthesis technique, the graphcut textures method [5] treats the problem of image texture synthesis as a min-cost graph cut problem, and employs graph cuts (e.g. [2, 8]) to remove (or reduce) the visible seams between patches so that the output results can be refined as required. In fact, graphcut textures is a graph-based texture synthesis technique.

Although the technique of graphcut textures has been successfully applied to image and video texture synthesis [5], there is a lack of a theoretical analysis of the technique, which is definitely desirable in both understanding and in using the technique for future research. We point out that it is not straightforward to give such a theoretical analysis. According to the general framework of graph cuts proposed by Boykov et al. [2, 8], depending on different problems (i.e. applications), each graph-based energy minimization method requires a different mathematical formulation of its graph in the sense that the energy function must be defined differently, and that the design of a proper energy function to encode the constraints of a problem is not trivial [2, 8]. In addition, the cuts and the min-cost cut of the graph have different mathematical properties for different problems, which play a crucial role in establishing the equivalence relationship (i.e. the key to solving a graph cut problem [2, 8]) between the set of certain type of cuts and the set of all labelings of the graph. Furthermore, special consideration must be taken into account in assigning a correct cost for each edge in the graph as suggested by Veksler in her Ph.D. thesis [8]. For example, in the $\alpha - expansion$ graph, when assigning a cost to an edge, it is stated that "We had to develop a special trick for the case when the original labels for p and q do not agree ($f_p \neq f_q$) in order to get the same effect that lemma 6 establishes for the simpler situation when $f_p = f_q$" [8].

More precisely, to give a theoretical analysis of graphcut textures, one has to address the following important issues: (1) what is the labeling problem of graphcut textures? Namely, what are the set of sites, the set of labels, the set of labelings, and the energy function of a labeling? (2) What type of graph is required for the labeling problem? (3) How to construct the graph? Namely, what are the vertices, the edges, and the costs of edges in the graph? (4) What mathematical properties a cut of the graph has? (5) What specific properties a min-cost cut has? (6) Given a cut, how to define its corresponding labeling? (7) Is it true that finding the optimal labeling is equivalent to finding the min-cost cut in the graph?

Except for the limited information provided on how to construct the graph, none of the above important theoretical issues has been addressed (or well addressed) in the paper of graphcut textures [5] or in any other publicly available publications that we are aware of.

Furthermore, there are some mathematical errors and confusions in the statements made regarding the graphs of graphcut textures (see the next section for the details) in the original paper, which must be clarified.

This paper addresses the above important issues in order to provide a complete and correct theoretical analysis of graphcut textures. The mathematical errors and confusions in Kwatra et al.'s original paper [5] are also clarified in this paper. To achieve this goal, we propose a new type of $\alpha-expansion$ graphs, called *complete* $\alpha-expansion$ graphs, which is an extension of the $\alpha-expansion$ graphs introduced by Boykov et al. [2, 8] . Using this new type of $\alpha-expansion$ graphs, all of the above important theoretical issues related to graphcut textures can be well formulated. It is shown that the objective of graphcut textures is to find an optimal labeling within a complete $\alpha-expansion$ move from the initial labeling f. The major theorem proved in this paper to support the graphcut textures is as follows: in a complete $\alpha-expansion$ graph, there is a one to one correspondence between the set of all elementary cuts and the set of all labelings within one complete $\alpha-expansion$ move of an initial labeling. This theorem indicates that finding the optimal labeling of graphcut textures is equivalent to finding the min-cost cut in a complete $\alpha-expansion$ graph, which can be efficiently computed by using graph cut techniques [1, 2, 4, 7, 8].

Another contribution of this paper is to provide a novel example of applying the general framework of the graph-based optimization methods proposed by Boykov, Veksler and Zabih [2, 8] to solve an application-dependent problem.

The rest of the paper is structured as follows. Section 2 provides the background knowledge on graph cuts and graphcut textures. Section 3 gives the mathematical analysis and proofs for graphcut textures using the complete $\alpha-expansion$ graphs. Finally, conclusions are drawn in Section 4.

2 Background Knowledge

2.1 Graph Cuts

In this paper, we use $\boldsymbol{G} =<\boldsymbol{V}, \boldsymbol{E}>$ to represent a weighted graph, where $\boldsymbol{V}$ and $\boldsymbol{E}$ are the set of vertices and the set of edges, respectively. There are two special vertices (called the *terminals*) in the graph: the *source* and the *sink*. A *cut* $\boldsymbol{C} \subset \boldsymbol{E}$ is a set of edges such that the terminals are separated in the induced graph $\boldsymbol{G}(\boldsymbol{C}) =<\boldsymbol{V}, \boldsymbol{E}-\boldsymbol{C}>$, and no proper subset of $\boldsymbol{C}$ separates the terminals in $\boldsymbol{G}(\boldsymbol{C})$. The cost of a cut $\boldsymbol{C}$, denoted by $|\boldsymbol{C}|$, is defined as the sum of its edge weights. A *min-cost cut* in the graph is a cut with the smallest cost, which can be efficiently computed using the max-flow theorem [4, 7].

Among the various optimization techniques of graph cuts [1], the methods introduced by Boykov, Vesler and Zabih [2, 8] have become popular for efficiently solving labeling problems in computer vision and computer graphics. In a labeling problem, a set of *sites* P and a set of *labels* L are given, the objective is to find the global (or nearly global) optimal labeling f (a *labeling* is a map from P to L) which minimizes the energy of f. In general, as indicated by Veksler in her Ph.D. thesis, "the design of a good energy function is not trivial" [8], and the energy function must properly incorporates the constraints of a problem [2, 8]. Another major difficulty in solving a labeling problem is that generally the problem is intractable. For example, for an image of size 32×32 (i.e. 1024 sites) with 256 gray levels (i.e. 256 labels) for each pixel, there are 256^{1024} labelings. Boykov, Vesler and Zabih [2, 8] have shown that graph cuts can be used to efficiently find the global or nearly global optimal solutions for labeling problems with energy functions that are everywhere smooth, piecewise constant, or piecewise smooth. The key idea of using graph cuts to solve a labeling problem is to construct a weighted graph in a way such that there is a one to one correspondence between the set of certain type of cuts in the graph and the set of all labelings that are in the move space of an initial labeling [2, 8]. With the one to one correspondence established, the labeling problem is converted to find the min-cost cut in the graph [2, 8]. However, both the construction of the graph and the establishment of the one to one correspondence are problem-dependent, and special tricks have to be developed as suggested by

Veksler [8]. In fact, there are similar difficult situations in graphcut textures [8], which are described below.

2.2 Graphcut Textures

The graphcut textures [5] is one of the state-of-art techniques in patch-based texture synthesis (e.g. [3, 6]). Given an input texture image, the patch-based texture synthesis generates an output texture image that is often larger than and looks similar to the input texture by copying small patches from the input to the output. Figure 1 gives an example of a simple patch-based texture synthesis, in which small patches are randomly chosen from the input and tiled together in the output. The problem of this simple method is that there are noticeable seams between the tiled patches (see Figure 1). In fact, the main goal of the patch-based texture synthesis is to find techniques to remove (or reduce) the visible seams between patches when they are placed into the output. In graphcut textures [5], the technique of graph cuts proposed by Boykov et al. [2] is used to remove the visible seams between patches.

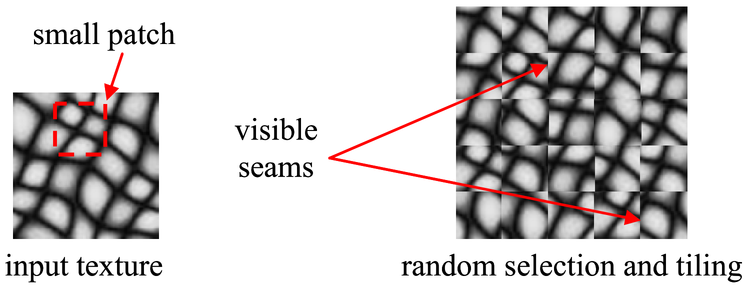


Figure 1: An example of simple patch-based texture synthesis.

Instead of tiling along their boundaries, patches are placed into the output with overlaps, i.e., a new patch is placed into the output such that it partially overlaps with existing (i.e. old) patches. To ensure that the textures in the newly placed patch maximally agree with those in the existing patches in the overlap region, a minimum error path in the overlap region (i.e. a path through the pixels in the overlap region where the new and old patches have the smallest intensity difference, see Figure 2 (a)) is calculated and used as a cut line between the new and old patches.

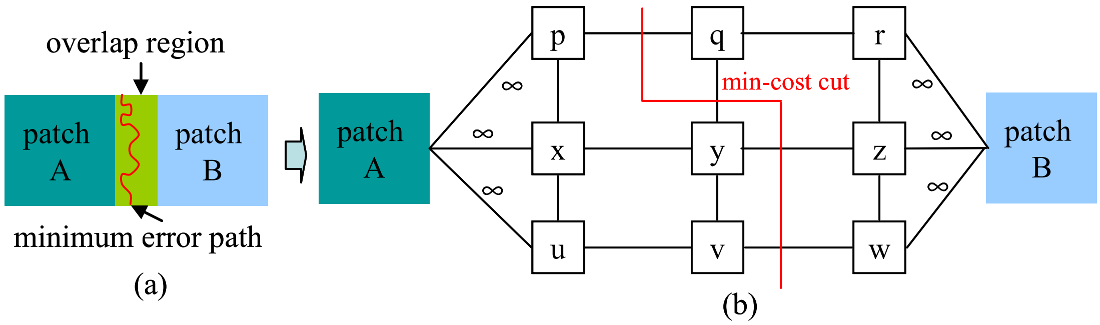


Figure 2: An example of graph formulation of finding the minimum error path.

The problem of finding the minimum error path in the overlap region can be considered as a graph cut problem. Suppose a new patch *B* is inserted into the output that overlaps with an existing patch *A*. A simple graph can be constructed as shown in Figure 2 (b). The existing patch in the output is represented by a source node with label "patch A" in the left, and the new patch is represented by a sink node with label "patch B" in the right. Both the source node and the sink node are called terminal nodes. For simplicity reasons, it is assumed that there are only 9 pixels in the overlap region between the new patch *B* and the existing patch *A*.

For each edge e_{pq} in the overlap region, where *p* and *q* are the neighboring nodes connected by the edge, a weighted cost $w(p,q,A,B)$ is assigned to it, which is defined as:

$$w(p,q,A,B) = | A(p) - B(p) | + | A(q) - B(q) |, \qquad (1)$$

where $A(p)$ is the gray level of pixel *p* from patch *A* and $B(p)$ is the gray level of *p* from patch *B*. If an edge between a terminal node (*A* or *B*) and a non-terminal node is assigned an infinite cost ∞, the non-terminal node will be forced to assume the label from the patch represented by the terminal node. For example, in Figure 2, both edges e_{Ap} and e_{zB} have ∞ costs, which imply that node *p* retains its old patch label and node *z* is assigned to the new patch *B*. The minimum error path in Figure 2 (a) is equivalent to the min-cost cut of the graph shown in Figure 2 (b), which can be calculated efficiently using standard graph cuts techniques (e.g. [4, 7]).

The above graph cut problem does not consider the visible seams between the old patches (called old seams) in the output. To remove (or reduce) the old seams when laying down a new patch into the output, a different type of graph is constructed as illustrated in Figure 3. The old patches, which are already placed down in the output, are represented by the source node with the label "existing patches A", and the new patch to be inserted is represented by the sink node with the label "new patch B". Let A_p represent the particular patch that a pixel p in the overlap region comes from. For each pair of neighboring pixels p and q in the overlap region, if $A_p \neq A_q$ (i.e. p and q come from different existing patches), then there is an old seam between p and q and a seam node s is created between p and q. Three edges e_{ps}, e_{sq} and e_{sB}, are also created at node s each with different weights assigned to them, which are $w(p,q,A_p,B)$, $w(p,q,B,A_q)$, and $w(p,q,A_p,A_q)$ (see Eq. 1 for the definition of $w(.)$), respectively. For the case of $A_p = A_q$, no information is provided in Kwatra's paper [5] . In fact, in this case, node p and q have the same initial patch label, and a weight of $w(p,q,A_p,B)$ should be assigned to edge e_{pq} as discussed in the next section.

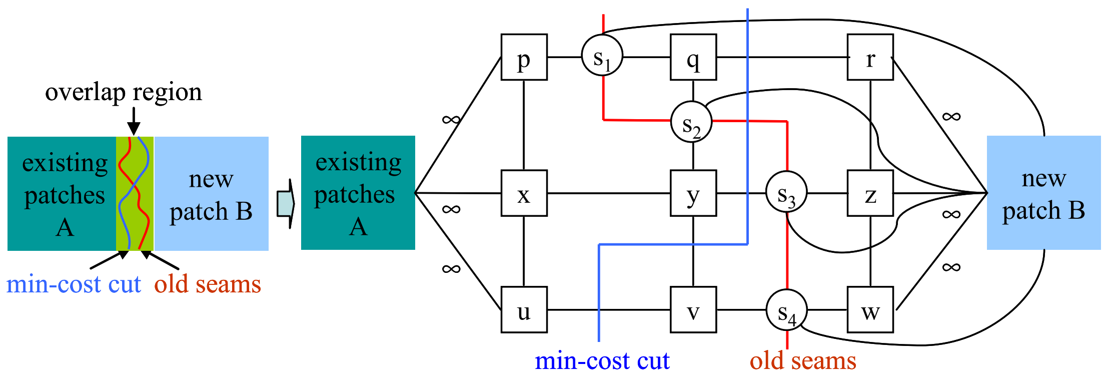


Figure 3: An example of graph with seam nodes added to incorporate old seam constraints.

For example, in Figure 3, there is an old seam between nodes p and q, then a seam node s_1 is created between them. In addition, the seam node s_1 is connected to the sink node B by a weighted edge, whose weight is $w(p,q,A_p,A_q)$. The edge between p and s_1 is assigned a weight

$w(p,q,A_p,B)$, and the edge between s_1 and q is assigned a weight $w(p,q,B,A_q)$. There is no old seam between node *x* and *y*, which implies *x* and *y* come from the same old patch, denoted by $A = A_x = A_y$, the weight for edge e_{xy} should be $w(x, y, A, B) = |A(x) - B(x)| + |A(y) - B(y)|$. Other seam nodes shown in Figure 3 are s_2, s_3 and s_4 between nodes *q* and *y*, *y* and *z*, and *v* and *w*, respectively.

Kwatra et al. [5] have argued that once the min-cost cut is calculated, some old seams (e.g. the one between node *v* and *w* in Figure 3) are removed from the graph. On the other hand, new seams (e.g. the ones between node *q* and *r*, *y* and *z*, *y* and *v*, and *u* and *v* in Figure 3) are also introduced into the graph by the min-cost cut. However, Kwatra et al. have not shown in their paper that the total seam cost (i.e. the overall intensity difference along the boundaries of patches) is reduced even with new seams introduced, which implies that the overall visible seams are reduced by the min-cost cut. Addressing the above problem requires the proof of the equivalence between the min-cost cut and the optimal labeling of the graph because the seam cost is represented as the energy of a labeling in the overlap region.

In addition, there are some mathematical errors in the statements made regarding the graphs constructed for graphcut textures (see the last paragraph in Section 3.1 of the original paper [5]), which are misleading. First, Kwatra et al. claim that there exists an equivalence relationship between the seam cost and the min-cost cut of the graph, which is inaccurate. In fact, as proved in Theorem 1 and Corollary 1 in the next section, there exists an equivalence relationship between the min-cost cut and the optimal labeling of the graph (not the seam cost), which indicates that the labeling problem in graphcut textures can be efficiently solved using existing graph cut techniques [1, 2, 4, 7, 8]. The relationship between the seam cost and the min-cost cut of the graph is not meaningful. Furthermore, there is no theoretical support to demonstrate that the proposed method can efficiently solve the labeling problem in graphcut textures using graph cuts.

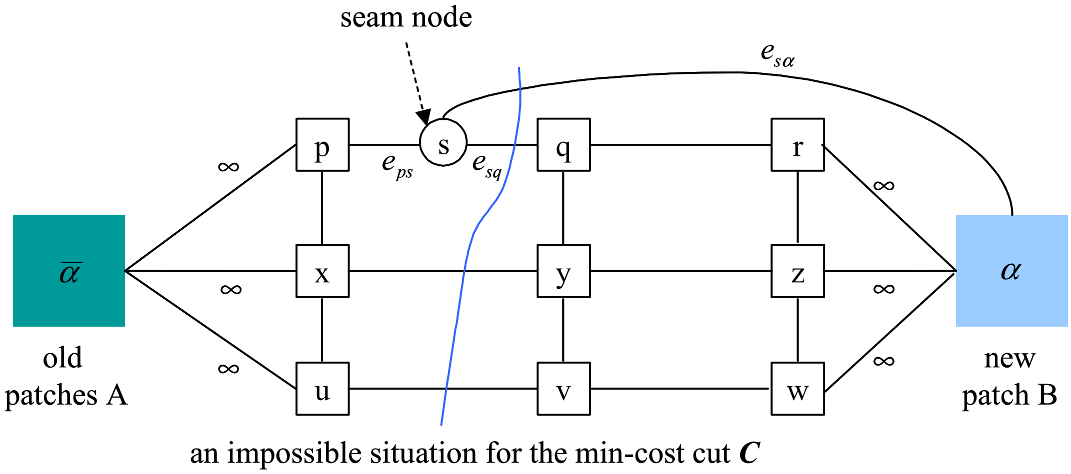


Figure 4: An example to demonstrate the situation when a min-cost cut cuts the edges at a seam node. It is impossible for the min-cost cut ***C*** to cut any two edges at a seam node.

Secondly, Kwatra et al. have made an incorrect comment on the min-cost cut which says that “picking two arcs originating from a seam node is always costlier than picking just one of them, hence at most one arc is picked in the min-cut”. The correct situation is that the min-cost cut can never pick two arcs from the same seam node, not to mention picking one of the two candidate arcs. As an example shown in Figure 4, the min-cost cut ***C*** cannot cut any two (e.g. e_{sq} and e_{sB}) of the three edges e_{ps}, e_{sq}, and e_{sB} at the seam node *s*. In fact, as proved in Lemma 3 and Lemma 4 in the next section, ***C*** can only cut either the single edge e_{ps}, or e_{sB}, or none of the three edges at *s*, and cannot even cut e_{sq}.

Lastly, the graph constructed for graphcut textures is not equivalent to the one used in Boykov et al.'s work [2] as claimed by Kwatra et al. in their paper [5]. By simply following Boykov's framework without reformulating the graph mathematically, one cannot prove the correctness of the graphcut textures, particularly the important equivalence relationship between the min-cost cut and the optimal labeling of the graph used in graphcut textures. In the next section, we show that the graph constructed as shown in Figure 3 is in fact a special type of $\alpha-expansion$ graph [2, 8], by which we call a *complete* $\alpha-expansion$ graph. We prove that there is a one to one correspondence between the set of all elementary cuts and the set of all labelings within one complete $\alpha-expansion$ move of the initial labeling *f* in the graph, and that the optimal labeling in graphcut textures is equivalent to the min-cost cut in the graph. The mathematical properties of the cuts, the min-cost cuts, the elementary cuts, and the corresponding labeling of a cut in a complete $\alpha-expansion$ graph are formulated in details in order to provide mathematical proofs for graphcut textures.

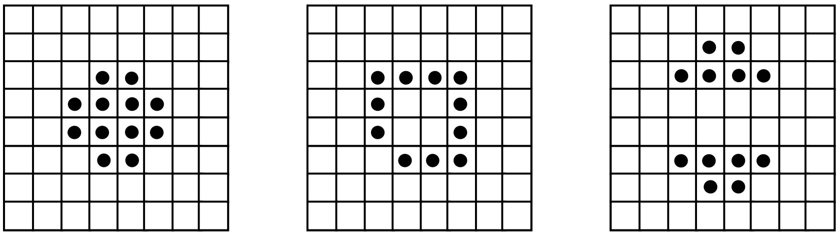
3 Mathematical Proofs

The $\alpha-$*expansion* graphs defined by Boykov et al. [2, 8] cannot be directly used to prove the correctness of the graphs constructed for graphcut textures. The reason is that the *expansion* space *A* in an $\alpha-$*expansion* move can be any subset of the set of sites *P*. However, in graphcut textures, each time when a new patch is laid down over existing patches in the output, a complete subregion of the new patch, where there are no holes, is used to replace a part of the overlap region. In fact, we need a new type of $\alpha-$*expansion* graphs that are constructed from *complete* $\alpha-$*expansion moves*, whose definition is given below. As shown later in this section, this particular type of $\alpha-$*expansion* moves plays an important role in defining the one to one correspondence between the set of elementary cuts and the set of all labelings of the graph constructed for graphcut textures. For clarity, the definition of $\alpha-$*expansion* move, which is taken from Boykov et al.'s work [2, 8], is restated first.

Definition 1 [2, 8] Let *P* be a set of sites, *L* be a set of labels, and ***F*** be the set of all labelings, where a *labeling f* is a map from *P* to *L*. Given a label $\alpha \in L$, a pair $(f, f') \in \boldsymbol{F} \times \boldsymbol{F}$ is an $\alpha-$*expansion move* if there exits $A \subset P$ such that $f'_p = \alpha$ for $p \in A$, and $f'_p = f_p$ for $p \notin A$.

One can see that f' can be obtained from f by switching all labels in A to α, and this means that the region of all sites with $\alpha-$labels has been expanded by f'. We call f' one $\alpha-$*expansion* move from f.

Definition 2 Without the loss of generality, assume four-nearest neighborhood connections are used. An $\alpha-$*expansion* move f' from f is *complete* if the set A is *complete* in the sense that both the set A and its complement $A^c = P \setminus A$ are connected under four-nearest neighborhood connections.



(a) (b) (c)

Figure 5: Examples of complete and non-complete sets. In all the figures, the set A is the set of all sites with black dots. The set A in (a) is complete, while the set A in either (b) or (c) is not complete. In (b), A is connected, but A^c is not. In (c), A^c is connected, but A is not.

A set A is *connected* if any two sites in it are connected by a path using four-nearest neighborhood connections. Examples of complete and non-complete sets are shown in Figure 5. The set A in Figure 5 (a) is complete since both A and A^c are connected. In Figure 5 (b), A is connected, but A^c is not, thus the set A is not complete. Similarly, the set A in Figure 5 (c) is not complete either. The complete set defined in Definition 2 naturally corresponds to the complete subregion (where there are no holes) of a new patch to be placed in the output image during the synthesis of graphcut textures.

With the complete $\alpha-expansion$ defined, the labeling problem in graphcut textures [5] can be stated as follows. Given a new patch α, a set of pixels P in the overlap region between the existing patches and the new patch, and an initial labeling f, the task is to find an optimal *complete* $\alpha-expansion$ *move* from f. We assume an initial labeling f as the one shown in Figure 3, where f always initializes some pixels in the overlap region (e.g. pixels r, z, and w in Figure 3) to the new patch. For any given labeling f, the energy of f, denoted by $E(f)$, is given by:

$$E(f) = \sum_{(p,q)\in \boldsymbol{N}} w(p,q,f_p,f_q), \qquad (2)$$

where f_p is the label of pixel p that tells which patch p comes from, $w(p,q,f_{p,}f_q)$ is defined as in Eq. 1, and $\boldsymbol{N}$ denotes the set of all neighboring pairs $\{p,q\}$ in the overlap region.

The energy defined in Eq. 2 measures the seam cost as described in the previous section. In fact, if two pixels p and q come from different patches (i.e. $f_p \neq f_q$), then there is a seam between them. In this case, we want to know how closely the textures in the two patches match with each other at pixel p and q. This can be measured by the color difference between p and q, which is exactly $w(p,q,f_p,f_q)$ according to Eq. 1 (note that a zero value of $w(p,q,f_p,f_q)$

means no difference in color, and thus it is a perfect match). If p and q come from the same patch (i.e. $f_p = f_q$), then these is no seam between them, which implies that the measure $w(p,q,f_p,f_q)$ should give a zero value. This is actually true by Eq. 1 and $f_p = f_q$.

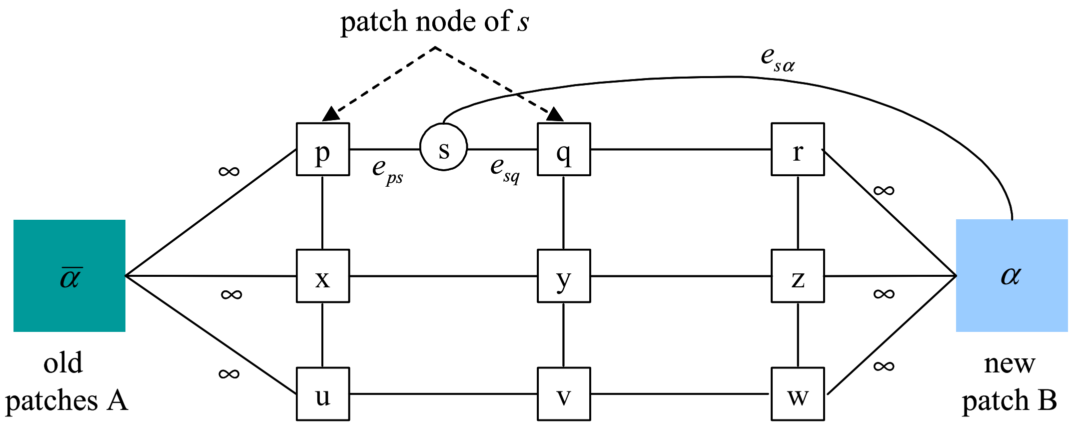


Figure 6: The structure of the graph built for graphcut textures.

The structure of the graph $\boldsymbol{G} = <\boldsymbol{V}, \boldsymbol{E}>$ is shown in Figure 6. The set of vertices includes the terminals α and $\overline{\alpha}$, as well as all the pixels P in the overlap region, where α represents the label for the new patch and $\overline{\alpha}$ stands for the old labels assigned to pixels in the overlap region by the initial labeling f. For each pair of neighboring pixels p and q, if they have different labels (i.e. $f_p \neq f_q$), then an *auxiliary* node [2, 8] (called the *seam* node in graphcut textures) $s = s_{pq}$ is created, and we call pixels p and q the *patch nodes* of the seam node s. Therefore, the set of vertices is

$$\boldsymbol{V} = \{\alpha\} \cup \{\overline{\alpha}\} \cup P \cup \left(\bigcup_{\substack{\{p,q\} \in N \\ f_p \neq f_q}} s_{pq} \right). \qquad (3)$$

An edge that connects a pixel $p \in P$ to one of the terminal nodes is called a *t-link* [2, 8] (e.g. edges $e_{\bar{\alpha}p}$ and $e_{r\alpha}$ in Figure 6) and the set of all *t*-links is represented by $\boldsymbol{T}$. For each pair of neighboring pixels $\{p,q\} \in \boldsymbol{N}$ with $f_p = f_q$, we connect them by an edge e_{pq}, which is called an *n-link* [2, 8]. For each pair of neighboring pixels $\{p,q\} \in \boldsymbol{N}$ with $f_p \neq f_q$, a triple of edges is created:

$$\boldsymbol{E}_{pq} = \{e_{ps}, e_{sq}, e_{s\alpha}\}, \tag{4}$$

where *s* is the seam node between *p* and *q*, the edge e_{ps} connects *p* to *s*, e_{sq} connects *s* to *q*, and $e_{s\alpha}$ connects *s* to α. Edges in $\boldsymbol{E}_{pq} = \{e_{ps}, e_{sq}, e_{s\alpha}\}$ are called *s-links*. Thus the set of all edges in the graph $\boldsymbol{G} = <\boldsymbol{V}, \boldsymbol{E}>$ is:

$$\boldsymbol{E} = \boldsymbol{T} \cup \left(\bigcup_{\substack{\{p,q\}\in N \\ f_p \neq f_q}} \boldsymbol{E}_{pq} \right) \cup \left(\bigcup_{\substack{\{p,q\}\in N \\ f_p = f_q}} \{e_{pq}\} \right). \tag{5}$$

Table 1: The weights assigned to edges in the graph shown in Figure 6.

Edge	Weight	Comments
$t \in \boldsymbol{T}$	∞	*t*-link
e_{ps}	$w(p,q,f_p,\alpha)$	$\{p,q\} \in \boldsymbol{N}, f_p \neq f_q$, *s* is the seam node between *p* and *q*, and α is the label for the new patch.
e_{sq}	$w(p,q,\alpha,f_q)$	
$e_{s\alpha}$	$w(p,q,f_p,f_q)$	
e_{pq}	$w(p,q,f_p,\alpha)$	$\{p,q\} \in \boldsymbol{N}, f_p = f_q$

By the requirement of the graphcut texture synthesis process [5], each *t*-link from a pixel $p \in P$ is assigned a weight of infinity, which prevents pixel *p*'s label from changing during complete $\alpha-$*expansion* moves (see Definition 2). The weights of *n*-links and *t*-links in the graph are assigned such that the results in Lemma 4 - Lemma 7 can hold, which are essential for the proofs of Theorem 1 and Corollary 1 described later. The weights for all edges in the graph are summarized in Table 1.

Lemma 1 Let s be a seam node in $\boldsymbol{G} = <\boldsymbol{V}, \boldsymbol{E}>$, and p and q be the patch nodes of s. For any cut $\boldsymbol{C}$ in $\boldsymbol{G} = <\boldsymbol{V}, \boldsymbol{E}>$, $e_{sq} \in \boldsymbol{C} \Rightarrow e_{s\alpha} \in \boldsymbol{C}$.

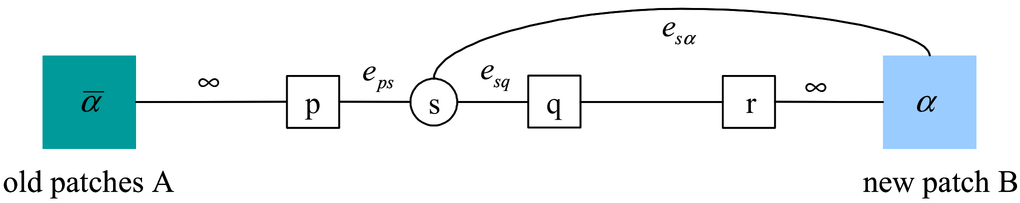


Figure 7: The smallest subgraph of the graph shown in Figure 6 that contains the terminals, the node p, s, and q, and all the corresponding edges among the nodes.

Proof: Let $\boldsymbol{G}_1 = <\boldsymbol{V}_1, \boldsymbol{E}_1>$ be the smallest subgraph of $\boldsymbol{G} = <\boldsymbol{V}, \boldsymbol{E}>$ that contains the terminals $\bar{\alpha}$ and α, the node p, s, and q, and all the corresponding edges among the nodes (see Figure 7). Now suppose $e_{sq} \in \boldsymbol{C}$, then $\boldsymbol{C}$ cannot contain e_{ps}. Otherwise, the cut $\boldsymbol{C}$ has a proper subset $\boldsymbol{C}_1 = \{e_{ps}\} \cup (\boldsymbol{C} \setminus \boldsymbol{E}_1)$ that separates the terminals in the induced graph $\boldsymbol{G}(\boldsymbol{C})$, which is a contradiction to the definition of a cut (see Section 2.1). On the other hand, since $\boldsymbol{C}$ is a cut in $\boldsymbol{G}$, $\boldsymbol{C}$ must cut each path between the terminals, which implies that $e_{s\alpha} \in \boldsymbol{C}$. Thus, we have proved that $e_{sq} \in \boldsymbol{C} \Rightarrow e_{s\alpha} \in \boldsymbol{C}$. ■

Lemma 2 Given a seam node *s* in $\boldsymbol{G} = <\boldsymbol{V}, \boldsymbol{E}>$. Let *p* and *q* be its patch nodes, and $\boldsymbol{E}_{pq}$ be the set of *s*-links at *s*. Any cut $\boldsymbol{C}$ in $\boldsymbol{G} = <\boldsymbol{V}, \boldsymbol{E}>$ satisfies exactly one of the three properties: (1) $\boldsymbol{C} \cap \boldsymbol{E}_{pq} = \phi$, (2) $\boldsymbol{C} \cap \boldsymbol{E}_{pq} = \{e_{ps}\}$, and (3) If $e_{sq} \in \boldsymbol{C}$ then $\boldsymbol{C} \cap \boldsymbol{E}_{pq} = \{e_{sq}, e_{s\alpha}\}$, else $\boldsymbol{C} \cap \boldsymbol{E}_{pq} = \{e_{s\alpha}\}$.

Proof: Let $\boldsymbol{G}_1 = <\boldsymbol{V}_1, \boldsymbol{E}_1>$ is the subgraph of $\boldsymbol{G}$ as defined in the proof of Lemma 1 (see Figure 7). Suppose $\boldsymbol{C} \cap \boldsymbol{E}_{pq} \neq \phi$, then at least one edge in $\boldsymbol{E}_{pq} = \{e_{ps}, e_{sq}, e_{s\alpha}\}$ must be in $\boldsymbol{C}$. If $e_{ps} \in \boldsymbol{C}$, then $\boldsymbol{C}$ cannot contain any of e_{sq} and $e_{s\alpha}$, and thus $\boldsymbol{C} \cap \boldsymbol{E}_{pq} = \{e_{ps}\}$. Otherwise there exists a proper subset $\boldsymbol{C}_1 = \{e_{ps}\} \cup (\boldsymbol{C} \setminus \boldsymbol{E}_1)$ of $\boldsymbol{C}$ that separates the terminals in the induced graph $\boldsymbol{G}(\boldsymbol{C})$, which contradicts the definition of a cut. If none of properties (1) and (2) is satisfied by $\boldsymbol{C}$, then the last property must hold for $\boldsymbol{C}$ according to Lemma 1. ■

Given an initial labeling *f* for the pixels in the overlap region, the objective is to find an optimal complete $\alpha-$*expansion* move from *f*. Now we prove that the optimal complete $\alpha-$*expansion* move is just the min-cost cut in the graph $\boldsymbol{G} = <\boldsymbol{V}, \boldsymbol{E}>$. Before proving this, we give the definition of an elementary cut in a graph.

Definition 3 An *elementary cut* $\boldsymbol{C}$ in $\boldsymbol{G} = <\boldsymbol{V}, \boldsymbol{E}>$ is a cut that cuts at most one of the three *s*-links in $\boldsymbol{E}_{pq} = \{e_{ps}, e_{sq}, e_{s\alpha}\}$ at any seam node *s*.

Lemma 3 Let s be a seam node in $\boldsymbol{G} = <\boldsymbol{V}, \boldsymbol{E}>$, p and q be its patch nodes, and $\boldsymbol{E}_{pq} = \{e_{ps}, e_{sq}, e_{s\alpha}\}$ be the set of s-links at s. Any elementary cut $\boldsymbol{C}$ in $\boldsymbol{G}$ satisfies exactly one of the three properties: (1) $\boldsymbol{C} \cap \boldsymbol{E}_{pq} = \phi$, (2) $\boldsymbol{C} \cap \boldsymbol{E}_{pq} = \{e_{ps}\}$, and (3) $\boldsymbol{C} \cap \boldsymbol{E}_{pq} = \{e_{s\alpha}\}$. Thus, any elementary cut $\boldsymbol{C}$ cannot contain the s-link e_{sq} in $\boldsymbol{E}_{pq} = \{e_{ps}, e_{sq}, e_{s\alpha}\}$ at any seam node s.

Proof: The results immediately follow from Lemma 2 and Definition 3. ■

Lemma 4 The min-cost cut $\boldsymbol{C}$ in $\boldsymbol{G} = <\boldsymbol{V}, \boldsymbol{E}>$ is an elementary cut.

Proof: By Lemma 2, as a cut, the min-cost cut $\boldsymbol{C}$ satisfies exactly one of three properties: (1) $\boldsymbol{C} \cap \boldsymbol{E}_{pq} = \phi$, (2) $\boldsymbol{C} \cap \boldsymbol{E}_{pq} = \{e_{ps}\}$, and (3) If $e_{sq} \in \boldsymbol{C}$ then $\boldsymbol{C} \cap \boldsymbol{E}_{pq} = \{e_{sq}, e_{s\alpha}\}$, else $\boldsymbol{C} \cap \boldsymbol{E}_{pq} = \{e_{s\alpha}\}$. To prove $\boldsymbol{C}$ is an elementary cut, we only have to show that if $\boldsymbol{C}$ satisfies property (3), then $\boldsymbol{C} \cap \boldsymbol{E}_{pq} = \{e_{s\alpha}\}$. Suppose this is not true (i.e. the "else" part does not hold for $\boldsymbol{C}$), then the "if" part holds for $\boldsymbol{C}$. Thus, we have $\boldsymbol{C} \cap \boldsymbol{E}_{pq} = \{e_{sq}, e_{s\alpha}\}$, and this is impossible by the minimality of the cost of $\boldsymbol{C}$ and by the fact that the costs (see Table 1) of the three edges in $\boldsymbol{E}_{pq}$ satisfies the triangle inequality (note that the cost function $w(.)$ defined in Eq. 1 is metric), which implies that cutting two edges e_{sq} and $e_{s\alpha}$ together in $\boldsymbol{E}_{pq}$ is costlier than cutting the third one (i.e. e_{ps}). ■

An elementary cut is not necessarily a min-cost cut. For example, in Figure 8, the cut $\boldsymbol{C}_1$ is an elementary cut, but not the min-cost cut, which is the cut $\boldsymbol{C}_2$. In order to establish a one to one correspondence between the set of all elementary cuts and the set of all labelings within one complete $\alpha - expansion$ move of the initial labeling *f*, we define the corresponding labeling of a cut as follows.

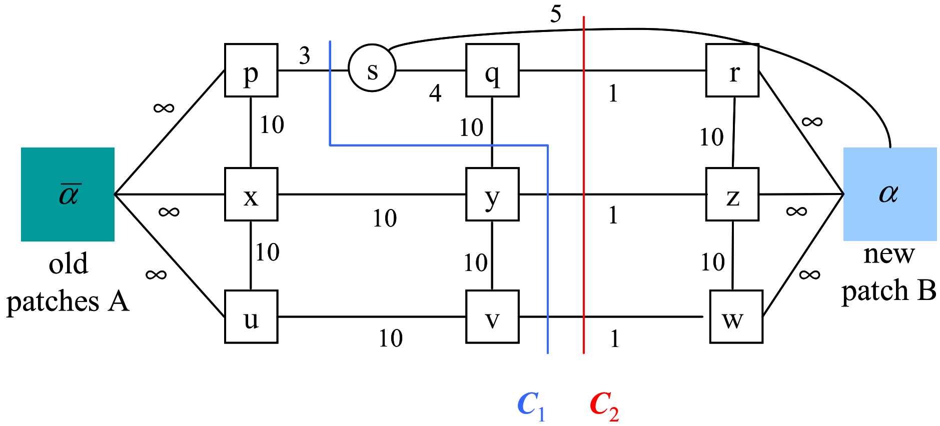


Figure 8: An example of an elementary cut C_1 that is not the min-cost cut.

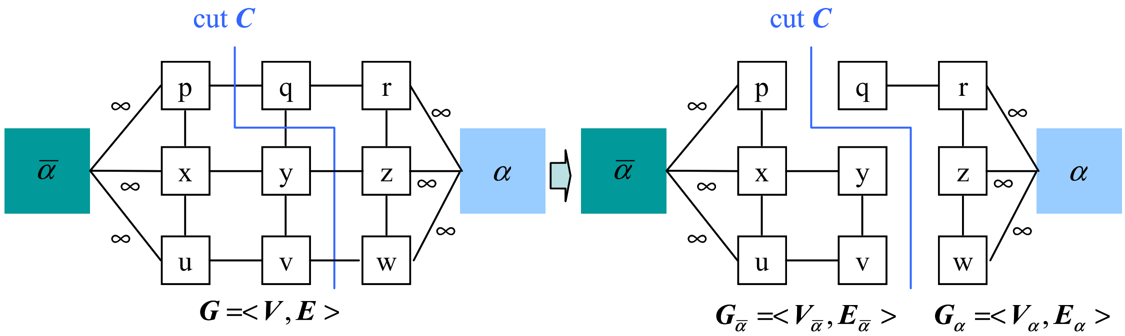


Figure 9: An example of two subgraphs $G_{\bar{\alpha}}$ and G_{α} of $G = < V, E >$ divided by cut C.

For a given cut C (not necessarily an elementary cut), let $G_{\bar{\alpha}} = < V_{\bar{\alpha}}, E_{\bar{\alpha}} >$ and $G_{\alpha} = < V_{\alpha}, E_{\alpha} >$ be the two subgraphs of G divided by C (i.e. the subgraphs after removing all the edges in $\boldsymbol{C}$ from $\boldsymbol{G}$, see Figure 9), the corresponding labeling $f^{\boldsymbol{C}}$ of $\boldsymbol{C}$ can be defined as follows:

$$f_p^{\boldsymbol{C}} = \begin{cases} f_p, & \text{if } p \in \boldsymbol{V}_{\bar{\alpha}} \\ \alpha, & \text{if } p \in \boldsymbol{V}_{\alpha} \end{cases}. \tag{6}$$

In other words, a pixel p is assigned label α if it belongs to the subgraph $\boldsymbol{G}_\alpha = <\boldsymbol{V}_\alpha, \boldsymbol{E}_\alpha>$, otherwise, it retains its old label.

Lemma 5 If $\{p,q\} \in \boldsymbol{N}$ ($\boldsymbol{N}$ is the set of all neighboring pairs) such that $f_p = f_q$, then any cut $\boldsymbol{C}$ in the graph $\boldsymbol{G} = <\boldsymbol{V}, \boldsymbol{E}>$ satisfies $|\boldsymbol{C} \cap \{e_{pq}\}| = w(p,q,f_p^{\boldsymbol{C}}, f_q^{\boldsymbol{C}})$, where the function $w(.)$ is defined in Eq. 1**.**

Proof: There are only two cases: (1) $\boldsymbol{C} \cap \{e_{pq}\} = \phi$ and (2) $\boldsymbol{C} \cap \{e_{pq}\} = \{e_{pq}\}$. Let $\boldsymbol{G}_{\bar{\alpha}} = <\boldsymbol{V}_{\bar{\alpha}}, \boldsymbol{E}_{\bar{\alpha}}>$ and $\boldsymbol{G}_\alpha = <\boldsymbol{V}_\alpha, \boldsymbol{E}_\alpha>$ be the two subgraphs divided by $\boldsymbol{C}$ as shown in Figure 9. If $\boldsymbol{C} \cap \{e_{pq}\} = \phi$, then p and q must belong to the same subgraph, i.e. either $\{p,q\} \in \boldsymbol{V}_{\bar{\alpha}}$ or $\{p,q\} \in \boldsymbol{V}_\alpha$. In either case, we have $f_p^{\boldsymbol{C}} = f_p = f_q = f_q^{\boldsymbol{C}}$ by Eq. 6, and this implies that $|\boldsymbol{C} \cap \{e_{pq}\}| = |\phi| = 0 = w(p,q,f_p,f_q) = w(p,q,f_p^{\boldsymbol{C}}, f_q^{\boldsymbol{C}})$ by Eq. 1. If $\boldsymbol{C} \cap \{e_{pq}\} = \{e_{pq}\}$, then p and q cannot belong to the same subgraph. Suppose $p \in \boldsymbol{V}_{\bar{\alpha}}$ and $q \in \boldsymbol{V}_\alpha$, then we have $f_p^{\boldsymbol{C}} = f_p$ and $f_q^{\boldsymbol{C}} = \alpha$. From the weights shown in Table 1, we have $|\boldsymbol{C} \cap \{e_{pq}\}| = |e_{pq}| = w(p,q,f_p,\alpha)$, which is equal to $w(p,q,f_p^{\boldsymbol{C}}, f_q^{\boldsymbol{C}})$ since $f_p^{\boldsymbol{C}} = f_p$ and $f_q^{\boldsymbol{C}} = \alpha$. ■

Lemma 6 If $\{p,q\} \in \boldsymbol{N}$ such that $f_p \neq f_q$, then any elementary cut $\boldsymbol{C}$ in the graph $\boldsymbol{G} =< \boldsymbol{V}, \boldsymbol{E} >$ satisfies $|\boldsymbol{C} \cap \boldsymbol{E}_{pq}| = w(p,q,f_p^{\boldsymbol{C}}, f_q^{\boldsymbol{C}})$, where $\boldsymbol{E}_{pq}$ is given in Eq. 4.

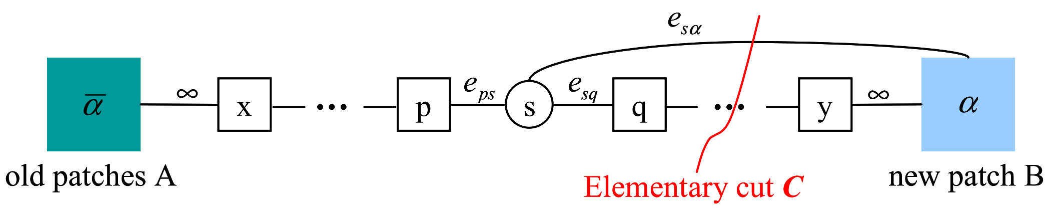


Figure 10: The smallest subgraph that contains terminal $\bar{\alpha}$ and α, the node *p*, *s*, and *q*, and all the corresponding edges among the nodes.

Proof: By Lemma 3, $\boldsymbol{C}$ satisfies exactly one of the three cases: (1) $\boldsymbol{C} \cap \boldsymbol{E}_{pq} = \phi$, (2) $\boldsymbol{C} \cap \boldsymbol{E}_{pq} = \{e_{ps}\}$, and (3) $\boldsymbol{C} \cap \boldsymbol{E}_{pq} = \{e_{s\alpha}\}$. We give the proof for case 3, and the proofs for the other two cases are similar. Suppose $\boldsymbol{C} \cap \boldsymbol{E}_{pq} = \{e_{s\alpha}\}$, let $\boldsymbol{G}_1 =< \boldsymbol{V}_1, \boldsymbol{E}_1 >$ be the smallest subgraph as shown in Figure 10 that contains terminal $\bar{\alpha}$ and α, the node *p*, *s*, and *q*, and all the corresponding edges among the nodes. One can see that $\boldsymbol{C}$ cuts all the paths between the terminals in $\boldsymbol{G}_1$. Let $\boldsymbol{G}_{\bar{\alpha}} =< \boldsymbol{V}_{\bar{\alpha}}, \boldsymbol{E}_{\bar{\alpha}} >$ and $\boldsymbol{G}_{\alpha} =< \boldsymbol{V}_{\alpha}, \boldsymbol{E}_{\alpha} >$ be the two subgraphs of $\boldsymbol{G}$ divided by $\boldsymbol{C}$ as shown in Figure 9. Since $\boldsymbol{C}$ is a cut and $\boldsymbol{C} \cap \boldsymbol{E}_{pq} = \{e_{s\alpha}\}$, it can only cut an edge after pixel *q* (see Figure 10), thus *p* and *q* must belong to the subgraph $\boldsymbol{G}_{\bar{\alpha}}$, i.e. $\{p,q\} \in \boldsymbol{V}_{\bar{\alpha}}$. By Eq. 6, we have $f_p^{\boldsymbol{C}} = f_p$ and $f_q^{\boldsymbol{C}} = f_q$. On the other hand, by Lemma 3 and Table 1, we have

$|\boldsymbol{C} \cap \boldsymbol{E}_{pq}| = |e_{s\alpha}| = w(p,q,f_p,f_q)$, which is equal to $w(p,q,f_p^{\boldsymbol{C}},f_q^{\boldsymbol{C}})$ since $f_p^{\boldsymbol{C}} = f_p$ and $f_q^{\boldsymbol{C}} = f_q$. ■

The results in Lemma 6 only hold for an elementary cut. For a non-elementary cut ***C***, it is possible that ***C*** contains both *s*-links e_{sq} and $e_{s\alpha}$ at a given seam node *s*. In that case, we have $|\boldsymbol{C} \cap \boldsymbol{E}_{pq}| \geq w(p,q,f_p^{\boldsymbol{C}},f_q^{\boldsymbol{C}})$ rather than $|C \cap \boldsymbol{E}_{pq}| = w(p,q,f_p^{\boldsymbol{C}},f_q^{\boldsymbol{C}})$.

Lemma 7 For any cut ***C*** in the graph $\boldsymbol{G} = <\boldsymbol{V},\boldsymbol{E}>$, its corresponding labeling $f^{\boldsymbol{C}}$ defined in Eq. 6 is a complete $\alpha - expansion$ move from the initial labeling *f*. Moreover, for any elementary cut ***C***, its cost is equal to the energy of $f^{\boldsymbol{C}}$, i.e.

$$|\boldsymbol{C}| = E(f^{\boldsymbol{C}}), \tag{7}$$

where $E(f^{\boldsymbol{C}})$ is given by Eq. 2.

Proof: It is obvious that $f^{\boldsymbol{C}}$ is an $\alpha - expansion$ move from the initial labeling *f*. As shown in Figure 9, both subregions $\boldsymbol{V}_\alpha$ and $\boldsymbol{V}_{\overline{\alpha}}$ are connected, thus by Definition 2, $f^{\boldsymbol{C}}$ is complete. Now, suppose ***C*** is an elementary cut, its cost $|\boldsymbol{C}|$ is calculated as follows:

$$|\boldsymbol{C}| = \sum_{\{p,q\}\in \boldsymbol{N}, f_p = f_q} |\boldsymbol{C} \cap \{e_{pq}\}| + \sum_{\{p,q\}\in \boldsymbol{N}, f_p \neq f_q} |\boldsymbol{C} \cap \boldsymbol{E}_{pq}| . \tag{8}$$

By Lemma 5, the first term in Eq. 8 is

$$\sum_{\{p,q\}\in \boldsymbol{N}, f_p = f_q} |\boldsymbol{C} \cap \{e_{pq}\}| = \sum_{\{p,q\}\in \boldsymbol{N}, f_p = f_q} w(p,q,f_p^{\boldsymbol{C}},f_q^{\boldsymbol{C}}).$$

By Lemma 6, the second term in Eq. 8 is

$$\sum_{\{p,q\}\in \boldsymbol{N}, f_p \neq f_q} |\boldsymbol{C} \cap \boldsymbol{E}_{pq}| = \sum_{\{p,q\}\in \boldsymbol{N}, f_p \neq f_q} w(p,q,f_p^{\boldsymbol{C}}, f_q^{\boldsymbol{C}}).$$

Plugging the above two equations into Eq. 8, we have

$$|\boldsymbol{C}| = \sum_{\{p,q\}\in \boldsymbol{N}, f_p = f_q} w(p,q,f_p^{\boldsymbol{C}}, f_q^{\boldsymbol{C}}) + \sum_{\{p,q\}\in \boldsymbol{N}, f_p \neq f_q} w(p,q,f_p^{\boldsymbol{C}}, f_q^{\boldsymbol{C}}) = \sum_{\{p,q\}\in \boldsymbol{N}} w(p,q,f_p^{\boldsymbol{C}}, f_q^{\boldsymbol{C}}) = E(f^{\boldsymbol{C}}). \blacksquare$$

Now, we present our main result in the following theorem, and give a corollary of the theorem which proves that the optimal complete $\alpha - expansion$ move f^* from an initial labeling *f* is the min-cost cut in the graph $\boldsymbol{G} = <\boldsymbol{V}, \boldsymbol{E}>$.

Theorem 1 There exists a one to one correspondence between the set of all elementary cuts in $\boldsymbol{G} = <\boldsymbol{V}, \boldsymbol{E}>$ and the set of all labelings within one complete $\alpha - expansion$ move of the initial labeling *f*. Moreover, for any elementary cut ***C***, the cost of ***C*** is equal to the energy of $f^{\boldsymbol{C}}$, i.e. $|\boldsymbol{C}| = E(f^{\boldsymbol{C}})$.

Proof: Let $\boldsymbol{\Omega}$ be the set of all elementary cuts in $\boldsymbol{G} = <\boldsymbol{V}, \boldsymbol{E}>$, and ***F*** be the set of all labelings within one complete $\alpha - expansion$ move of the initial labeling f. We define a map between $\boldsymbol{\Omega}$ and ***F*** as follows:

$$\begin{aligned} \varphi : \boldsymbol{\Omega} &\rightarrow \boldsymbol{F} \\ \boldsymbol{C} &\mapsto \varphi(\boldsymbol{C}) = f^{\boldsymbol{C}} \end{aligned} \qquad (9)$$

where $f^{\boldsymbol{C}}$ is defined in Eq. 6. We show that the map φ is a one to one correspondence, i.e. both an injection and surjection.

We first prove that φ is an injection, i.e. for any two distinct elementary cuts $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$, we want to show that their corresponding labelings $f_1 = f^{\boldsymbol{C}_1}$ and $f_2 = f^{\boldsymbol{C}_2}$ are different, i.e. $f_1 \neq f_2$. Let $\boldsymbol{G}_{\bar{\alpha}}(\boldsymbol{C}_1) = <\boldsymbol{V}_{\bar{\alpha}}(\boldsymbol{C}_1), \boldsymbol{E}_{\bar{\alpha}}(\boldsymbol{C}_1)>$ and $\boldsymbol{G}_{\alpha}(\boldsymbol{C}_1) = <\boldsymbol{V}_{\alpha}(\boldsymbol{C}_1), \boldsymbol{E}_{\alpha}(\boldsymbol{C}_1)>$ be the two subgraphs (see Figure 9 for an example) divided by $\boldsymbol{C}_1$, and $\boldsymbol{G}_{\bar{\alpha}}(\boldsymbol{C}_2) = <\boldsymbol{V}_{\bar{\alpha}}(\boldsymbol{C}_2), \boldsymbol{E}_{\bar{\alpha}}(\boldsymbol{C}_2)>$ and $\boldsymbol{G}_{\alpha}(\boldsymbol{C}_2) = <\boldsymbol{V}_{\alpha}(\boldsymbol{C}_2), \boldsymbol{E}_{\alpha}(\boldsymbol{C}_2)>$ be the two subgraphs divided by $\boldsymbol{C}_2$. Since $\boldsymbol{C}_1 \neq \boldsymbol{C}_2$, their corresponding subgraphs must be different, and this implies that $\boldsymbol{V}_{\alpha}(\boldsymbol{C}_1) \neq \boldsymbol{V}_{\alpha}(\boldsymbol{C}_2)$ and $\boldsymbol{V}_{\bar{\alpha}}(\boldsymbol{C}_1) \neq \boldsymbol{V}_{\bar{\alpha}}(\boldsymbol{C}_2)$. Therefore, there exists at least one pixel p such that $p \in \boldsymbol{V}_{\alpha}(\boldsymbol{C}_1)$ and $p \notin \boldsymbol{V}_{\alpha}(\boldsymbol{C}_2)$, which implies that $f_1(p) = \alpha \neq f_2(p)$. Thus, we have proved that $f_1 \neq f_2$.

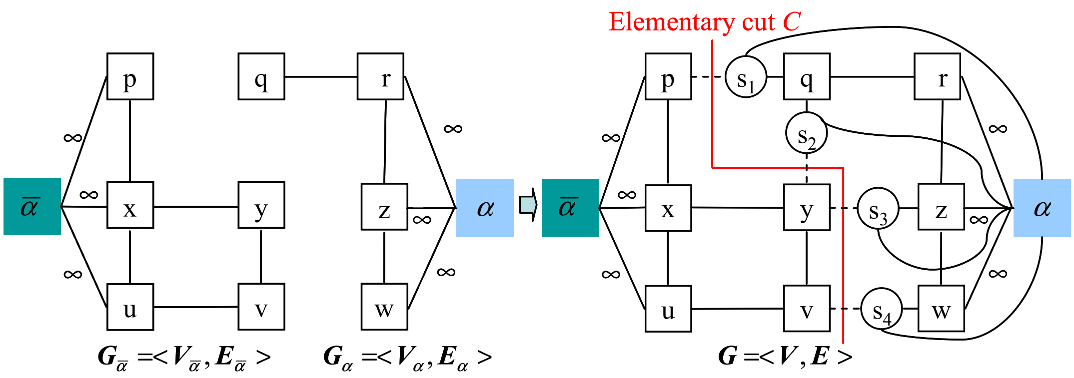


Figure 11: An example of how to construct an elementary cut ***C***, given a complete $\alpha - expansion$ f' from the initial labeling *f*.

To prove φ is a surjection, for any complete $\alpha - expansion$ f' from the initial labeling *f*, we need to show that there exists an elementary cut $\boldsymbol{C} \in \boldsymbol{\Omega}$ such that $\varphi(\boldsymbol{C}) = f'$. The elementary cut $\boldsymbol{C}$ can be constructed as follows. Since f' is a complete $\alpha - expansion$, so the set of all pixels with α labels are connected, and the set of all pixels with old labels (i.e. $\overline{\alpha}$) are also connected according to Definition 2. Let $\boldsymbol{G}_\alpha = <\boldsymbol{V}_\alpha, \boldsymbol{E}_\alpha>$ be the subgraph where all pixels have label α, and $\boldsymbol{G}_{\overline{\alpha}} = <\boldsymbol{V}_{\overline{\alpha}}, \boldsymbol{E}_{\overline{\alpha}}>$ be the subgraph where all pixels have old labels (see the left figure in Figure 11). For each pair of neighboring pixels $\{p,q\} \in \boldsymbol{N}$ where $p \in \boldsymbol{V}_{\overline{\alpha}}$ and $q \in \boldsymbol{V}_\alpha$, let *s* be the seam node between *p* and *q*, then we define the set of all edges e_{ps} as the required cut $\boldsymbol{C}$ (e.g. the set of all dashed lines in the right figure in Figure 11), i.e.

$$\boldsymbol{C} = \{e_{ps} \mid e_{ps} \in \boldsymbol{E}_{pq} = \{e_{ps}, e_{sq}, e_{s\alpha}\}, \{p,q\} \in \boldsymbol{N}, p \in \boldsymbol{V}_{\overline{\alpha}}, q \in \boldsymbol{V}_\alpha\}\,. \tag{10}$$

Clearly, the cut $\boldsymbol{C}$ defined in Eq. 10 is an elementary cut, and satisfies $f^{\boldsymbol{C}} = f'$, i.e. $\varphi(\boldsymbol{C}) = f'$. The "moreover" part of Theorem 1 has been proved in Lemma 7. ■

As discussed in the beginning of this section, the objective of graphcut textures is to find an optimal labeling within a complete $\alpha - expansion$ move from the initial labeling *f*. The following corollary indicates that finding the optimal labeling is equivalent to finding the min-cost cut in the graph, which can be efficiently computed by using graph cut techniques [2, 8].

Corollary 1 Let $\boldsymbol{C}$ be the min-cost cut in $\boldsymbol{G} = <\boldsymbol{V}, \boldsymbol{E}>$. Then the optimal labeling $f^* = \underset{f'}{arg}\ \min E(f')$ among f' is given by $f^* = f^{\boldsymbol{C}}$, where f' is a complete $\alpha - expansion$ move from the initial labeling *f*.

Proof: Let $\boldsymbol{F} = \{f' \mid f' \text{ is a complete } \alpha - \text{expansion from } f\}$, and $\boldsymbol{\Omega}$ be the set of all elementary cuts in $\boldsymbol{G}$. By Theorem 1, there exists a one to one correspondence between $\boldsymbol{G}$ and $\boldsymbol{F}$. Since $\boldsymbol{C}$ is the min-cost cut in $\boldsymbol{G}$, it is an elementary cut by Lemma 4, thus $\boldsymbol{C} \in \boldsymbol{\Omega}$. The cost of $\boldsymbol{C}$ is $|\boldsymbol{C}| = E(f^{\boldsymbol{C}})$ again by Theorem 1 . The corresponding labeling $f^{\boldsymbol{C}}$ of $\boldsymbol{C}$ is optimal since $\boldsymbol{C}$ is the min-cost cut, i.e. $f^* = \underset{f' \in \boldsymbol{F}}{arg} \ \min E(f') = f^{\boldsymbol{C}}$. ■

4 Conclusions

Although the technique of graphcut textures [5] has been successfully used in image and video texture synthesis since it was introduced in 2003, the mathematical formulation and mathematical properties of the graphs for the labeling problem of graphcut textures are not well presented in the original paper, which make the technique unconvincing from the theoretical point of view. This paper gives the essential mathematical support and proofs for graphcut textures, and clarifies the mathematical errors and confusions in the original graphcut textures paper. We prove that the labeling problem (i.e. finding the optimal labeling) of graphcut textures is equivalent to finding the min-cost cut in a complete $\alpha - expansion$ graph.

Acknowledgements

The authors would like to thank the anonymous reviewers and appropriate agencies.

References

1. Ahija, K., T.L. Magnati, and J.B. Orlin, *Network Flows: Theory, Algorithm, and Applications.* Prentice Hall, 1993.
2. Boykov, Y., O. Veksler, and R. Zabih, *Efficient Approximate Energy Minimization via Graph Cuts.* IEEE PAMI (also in ICCV 99), 2001. **23**(12): p. 1222-1239.
3. Efros, A.A. and W.T. Freeman, *Image Quilting for Texture Synthesis and Transfer.* ACM SIGGRAPH, 2001: p. 341-346.
4. Ford, L. and D. Fulkerson, *Flows in Networks.* Princeton University Press, 1962.
5. Kwatra, V., et al., *Graphcut Textures: Image and Video Synthesis using Graph Cuts.* ACM SIGGRAPH, 2003. **22**(3): p. 227-286.
6. Liang, L., et al., *Real-Time Texture Synthesis by Patch-Based Sampling.* ACM Transactions on Graphics, 2001. **20**(3): p. 127-150.
7. Sedgewick, R., *Algorithms in C, Part 5: Graph Algorithms.* Addison-Wesley, Reading, Massachusetts, 2001.
8. Veksler, O., *Efficient Graph-Based Energy Minimization Methods in Computer Vision.* Ph. D. Thesis, Cornell University, 1999.